

Ambito de decision (Objetos / Arquitectura / Persistencia / Otro)A:E	Componente/s impactado/s	Decisión	Otras Alternativas	Justificación de la decisión
Objetos	Documento	No hacer clase "Documento"	Vimos la posibilidad de crear la clase para tener por separado el tipo y numero de documento	No vimos necesario crear una clase aparte porque no vimos eficiente utilizacion futura
Objetos	Nombre de heladera	Que sea un atributo de la heladera	Que sea un atributo de la ubicación que se relaciona con la heladera por comprension del enunciado	Decidimos que quede como un atributo de la heladera ya que un nombre no lo consideramos como parte de la definicion de ubicación
Objetos	PersonaVulnerable	Dejar los atributos y si no los posee que se dejen en NULL	Pensamos en poner los booleanos como atributos (poseeDireccion/poseeDocumento) y solicitarlos en caso de True	Decidimos esto para simplificar los atributos y metodos de verificacion posteriores
FIN DE JUSTIFICACIONES - ENTREGA N°1				
Objetos	Documento	Hacer clase "Documento"	Vimos la posibilidad de que TipoDocumento sea un Enum	Vimos necesario crear una clase aparte para que los datos se puedan manejar de una manera más modularizada
Objetos	Heladera	Que sean un atributo de la heladera	Que sea un atributo de la ubicación que se relaciona con la heladera por comprension del enunciado	Decidimos que quede como un atributo de la heladera ya que un nombre no lo consideramos como parte de la definicion de ubicación
Objetos	Heladera	Una heladera debe poder tener su propia temperatura maxima y temperatura minima	otra posible solucion pudo haber sido que el receptor propio de la heladera tenga su temperatura minima y maxima, ya que es ella quien va a evaluarla.	decidimos ponerlo en heladera ya que es un elemento propio de cada heladera
Objetos	PersonaVulnerable	Dejar los atributos y si no los posee que se dejen en NULL	Pensamos en poner los booleanos como atributos (poseeDireccion/poseeDocumento) y solicitarlos en caso de True	Decidimos esto para simplificar los atributos y metodos de verificacion posteriores
Objetos	Colaborador	Que la relación entre Colaborador y cada tipo de Colaboracion, se delegue en la parte del colaborador. Que el Colaborador chequee qué Colaboraciones puede hacer.	Relacionar c/Colaborador con c/tipo de Colaboracion que pueden hacer	Para simplificacion y mejor trato del diagrama
Objetos	Colaboración	Decidimos tener una lista que contenga los tipos de personas habilitadas para realizar cada colaboración en específico	Otra alternativa que planteamos fue que cada colaborador tenga una lista de colaboraciones habilitadas	Mejor delegación. Es una mala práctica por si queremos agregar mas tipos de colaboraciones en el futuro, no haya que modificar a las personas.

Objetos	Sensores	Que la heladera tenga receptores los cuales reciben un sensorio (el de temperatura) y una alerta (el de movimiento)	Los receptores conocian a la heladera y los sensores conocian (utilizaban) a los receptores	Para simplificación del diagrama y para que no haya tantas clases que no cumplan ninguna funcionalidad
Objetos	Técnico	Decidimos hacer una clase para el Técnico, y que este también utilice el Enum de MedioContacto	Otra alternativa posible sería realizar una clase Persona, y que de esta se herede Colaborador y Tecnico.	No optamos por realizar la alternativa ya que no presentan muchos atributos en común.
Objetos	PersonaVulnerable	Como se quiere conocer en todo momento quien dio de alta a una persona vulnerable, decidimos relacionarle un atributo con su PersonaHumana que la dio de alta	Otra posible alternativa seria que la PersonaHumana disponga una lista con todas las PersonasVulnerables a las que dio de alta	Conocer en todo momento quien dio de alta una PersonaVulnerable
Objetos	Colaboracion	Decidimos por conveniencia enviar al Colaborador por parametro cuando se realiza una nueva colaboracion para poder realizar la suma de puntos desde la propia colaboracion	Otra alternativa podria ser tener una clase que se encargue de calcular los puntos cada que el Colaborador requiera conocerlos, tanto para canjearlos o simplemente para conocer su puntaje actual	Decidimos tomar esta forma, ya que notamos que era poco performante tener una clase que calcule este dato, pudiendo tener un Colaborador con un gran numero de Colaboraciones, sin poder identificar en cual se utilizaron los puntos.
Objetos	Rubro de oferta	Decidimos realizar un Enum que contenga los diferentes rubros de ofertas.	Otra alternativa podría ser que el rubro sea un atributo dentro de cada oferta.	Tomamos esta decisión para tener mejor delegación, en caso de que en un futuro se quieran agregar más rubros. Además decidimos esto ya que la oferta dependerá del rubro.
Objetos	PersonaJuridica	Decidimos implementar una lista de heladeras en PersonaJuridica.	-	Para poder conocer el puntaje que genera cada heladera activa, decidimos justamente generar una cronetask para conocer el puntaje que estas generan. También lo decidimos ya que queremos conocer mensualmente cuanto puntaje se le tiene
Objetos	RegistroPersonasSituVulnerable	Para conocer el puntaje que suma entregar tarjetasPlasticas, decidimos que a la hora de entregar una tarjeta, automaticamente se llame a un metodo que sume los puntos pertinentes respecto a la entrega de esa tarjeta.	-	Decidimos realizarlo de esta forma, ya que no sabemos de antemano cuantas tarjetas repartio, y en que momento.
Objetos	Colaboracion	El puntaje se calcula dentro de la misma colaboración	Hacer un calculador de puntaje el cual era una interfaz	Decidimos esto ya que no podemos conocer de antemano cuantos puntos va a generar una colaboracion, como lo es registrar personas vulnerables, por lo que, implementamos un metodo donde cada colaboracion se encargue de
Objetos	Ubicacion	Clase aparte llamada ubicacion la cual contiene latitud, longitud y direccion como atributos	-	Dedicimos esto para mejor manejo de las clases y de la API
Objetos	Validador	Decidimos implementar una interfaz llamada Condición	-	Decidimos esto ya que si se quieren agregar más condiciones, se pueden agregar sin necesidad de modificar lo demás, y la validación de contraseña se hará en base a la cantidad de validaciones que haya.

Objetos	Notificador	Decidimos implementar un conjunto de patrones en este paquete, tanto Strategy como Adapters	-	Esta decision fue tomada a base de poder ampliar facilmente la forma de enviar notificaciones, tanto si se quisiese agregar un metodo nuevo en la forma de notificar, como la forma propia de notificar de un metodo.
FIN DE JUSTIFICACIONES - ENTREGA N°2				
Objetos	Distribución de viandas	Al momento de distribuir viandas por un desperfecto en la heladera origen, se fija el espacio que tiene disponible y distribuye en esa heladera la mayor cantidad de viandas posible.	-	Realizamos esta implementación para que, en caso de que no haya más lugar en la heladera destino, no realizará la distribución de viandas.
Objetos	Alertas	Decidimos no crear una clase "Falla de conexión" ya que cuando lee el sensorio, verifica que este tenga fecha y hora distinta al anterior levantado. De ser el caso se reporta un error en el sensorio de temperatura.	Realizar una clase abstracta "Alerta" de la cual hereden los 3 tipos de alertas	Decidimos realizar un Enum con los tipos de alertas ya que estos tipos no tenían comportamiento.
Objetos				